

**What is claimed is:**

1. A method comprising:  
generating an instrumented code of the application;  
executing a plurality of tests on the instrumented code of the application;  
generating one or more test profiles associated with the plurality of tests; and  
identifying at least one of the plurality of tests based on the one or more test profiles.
2. A method as defined in claim 1, wherein generating the instrumented code of the application comprises inserting one or more probes into the application.
3. A method as defined in claim 1, wherein generating the one or more test profiles associated with the plurality of tests comprises identifying one or more program states of the application and generating one or more time stamps corresponding to each of the one or more program states.
4. A method as defined in claim 1, wherein generating the one or more test profiles associated with the plurality of tests comprises generating one or more time stamps indicative of an earliest time corresponding to a breakpoint of the application associated with one or more program states.
5. A method as defined in claim 1, wherein generating the one or more test profiles associated with the plurality of tests comprises generating one or more time stamps corresponding to one or more program states of the application based on at least one of a hardware timer, a software timer, and a virtual timer.

6. A method as defined in claim 1, wherein identifying the at least one of the plurality of tests based on the one or more test profiles comprises generating a priority list having the at least one of the plurality of tests to identify one or more breakpoints of the application associated with one or more program states.

7. A method as defined in claim 1, wherein identifying the at least one of the plurality of tests based on the one or more test profiles comprises identifying the at least one of the plurality of tests based on the one or more test profiles in response to a query.

8. A method as defined in claim 1 further comprising storing the one or more test profiles in a database.

9. A machine readable medium storing instructions that, when executed, cause a machine to:

generate an instrumented code of the application;

execute a plurality of tests on the instrumented code of the application;

generate one or more test profiles associated with the plurality of tests; and

identify at least one of the plurality of tests based on the one or more test profiles.

10. A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to generate the instrumented code of the application by inserting one or more probes into the application.

11. A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to generate the one or more test profiles associated with the plurality of tests by identifying one or more program states of the application and generating one or more time stamps corresponding to each of the one or more program states.

12. A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to generate the one or more test profiles associated with the plurality of tests by generating one or more time stamps indicative of an earliest time corresponding to a breakpoint of the application associated with one or more program states.

13. A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to generate the one or more test profiles associated with the plurality of tests by generating one or more time stamps corresponding to one or more program states of the application based on at least one of a hardware timer, a software timer, and a virtual timer.

14. A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to identify the at least one of the plurality of tests based on the one or more test profiles by generating a priority list having the at least one of the plurality of tests to identify one or more breakpoints of the application associated with one or more program states.

15. A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to identify the at least one of the plurality of tests based on the one or more test profiles and user input by generating a priority list having at least one of the plurality of tests to identify the one or more program states for each breakpoint of the application.

16. A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to identify the at least one of the plurality of tests based on the one or more test profiles comprises identifying at least one of the plurality of tests based on the one or more test profiles in response to a query.

17. A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to store the one or more test profiles in a database.

18. A machine readable medium as defined in claim 9, wherein the machine readable medium comprises at least one of a programmable gate array, application specific integrated circuit, erasable programmable read only memory, read only memory, random access memory, magnetic media, and optical media.

19. An apparatus comprising:  
a data structure configured to store one or more test profiles;  
a code coverage device configured to generate an instrumented code of the application;

a debugging and testing device configured to execute a plurality of tests on the instrumented code of the application, and to generate one or more test profiles associated with the plurality of tests; and

a test identifying device configured to identify at least one of the plurality of tests based on the one or more test profiles.

20. An apparatus as defined in claim 19, wherein the code coverage device comprises at least one of a compiler, an assembler, an interpreter, and a post-link optimizer.

21. An apparatus as defined in claim 19, wherein the instrumented code of the application comprises one or more probes to identify one or more program states of the application and to generate one or more time stamps corresponding to each of the one or more program states.

22. An apparatus as defined in claim 19, wherein the one or more test profiles comprises one or more time stamps indicative of an earliest time corresponding to a breakpoint of the application associated with one or more program states of the application.

23. An apparatus as defined in claim 19, wherein the test identifying device is configured to identify at least one of the plurality of tests based on the one or more test profiles in response to a query.

24. An apparatus as defined in claim 19, further comprising a test prioritizing device configured to generate a priority list having at least one of the plurality of tests.

25. A processor system comprising:  
a dynamic random access memory (DRAM) configured to store at least one test and;  
a processor operatively coupled to the DRAM, the processor being configured to:  
generate an instrumented code of the application;  
execute a plurality of tests on the instrumented code of the application;  
generate one or more test profiles associated with the plurality of tests; and  
identify at least one of the plurality of tests based on the one or more test profiles.

26. A processor system as defined in claim 25, wherein the instrumented code of the application comprises one or more probes inserted into the application to identify one or more program states of the application and to generate one or more time stamps corresponding to each of the one or more program states.

27. A processor system as defined in claim 25, wherein the one or more test profiles comprises one or more time stamps corresponding to one or more program states of the application.

28. A processor system as defined in claim 25, wherein the one or more test profiles comprises one or more time stamps indicative of an earliest time corresponding to a breakpoint of the application associated with one or more program states of the application.

29. A processor system as defined in claim 25, wherein the processor is configured to generate a priority list having the at least one of the plurality of tests to identify one or more breakpoints of the application associated with one or more program states.

30. A processor system as defined in claim 25, wherein the processor is configured to identify the at least one of the plurality of tests based on the one or more test profiles in response to a query.